Power and Energy Normalized Speedup Models for Heterogeneous Many Core Computing

Mohammed A. N. Al-hayanni[1], Ashur Rafiev[2], Rishad Shafik[1], **Fei Xia[1]**
School of EEE[1] and CS[2], Newcastle University
Newcastle Upon Tyne, NE1 7RU, UK

Outline

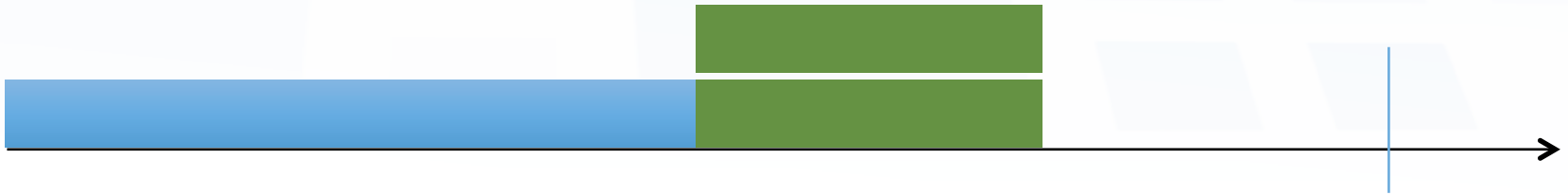
- Existing speedup models
- Motivation
- Extended heterogeneous speedup models
- Power consumption models
- Power and energy normalized speedup
- Experimental results and cross validation
- Conclusions

Amdahl's Law

- Fixed workload
  - (50% parallelizable P=0.5)
  - On a sequential processor (single core) takes 1 unit of time to complete

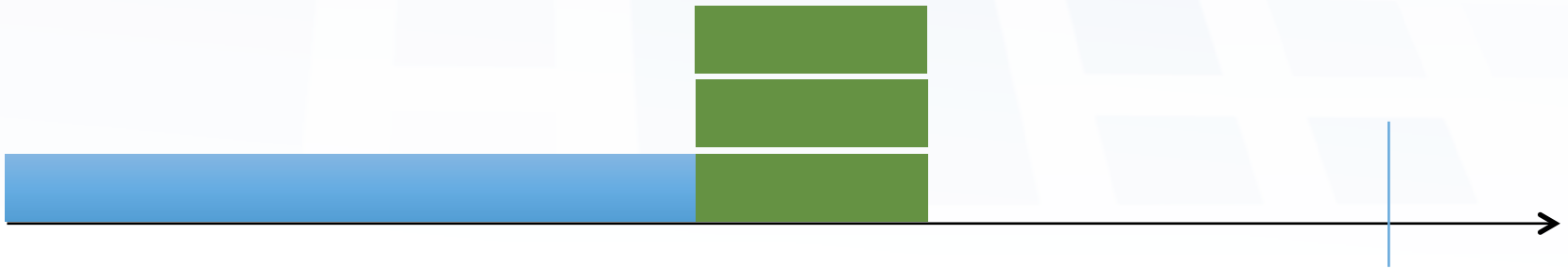
Amdahl's Law

- With two cores …
  - Parallelizable part is distributed between the two cores
  - Total time 0.75
  - Speedup = 1/0.75 = 1.333



Amdahl's Law

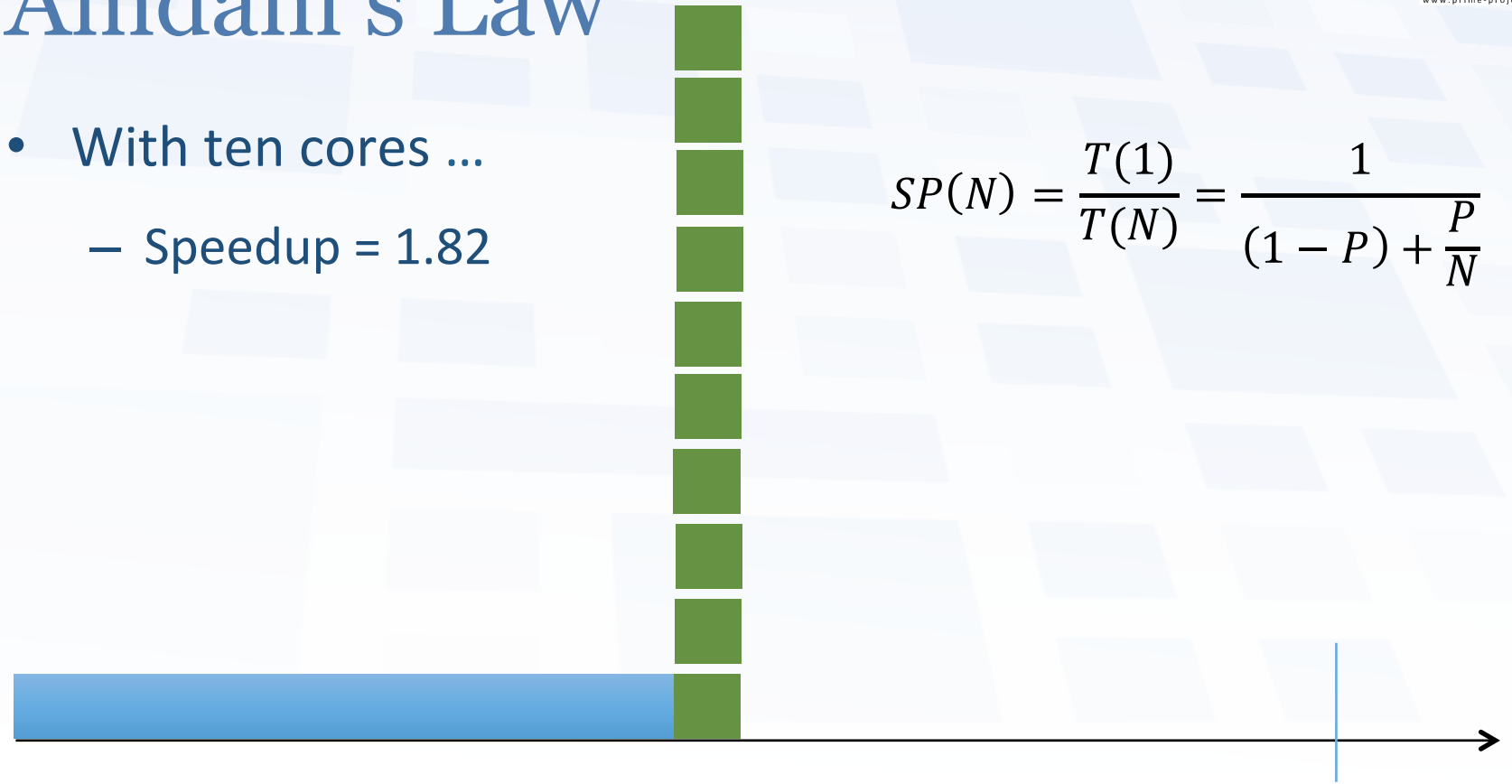
- With three cores …
  - Speedup = 1/0.667 = 1.5



Amdahl's Law

- With ten cores …
  - Speedup = 1.82

$$SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1-P) + \frac{P}{N}}$$



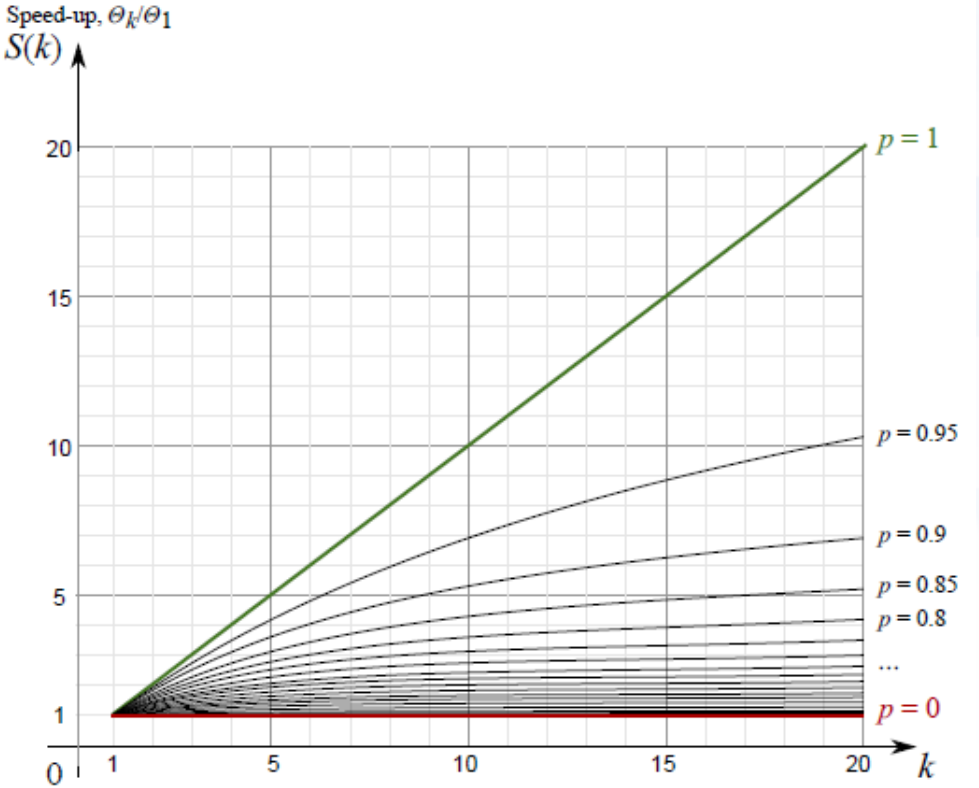
Amdahl's Law

- With ∞ cores …
  - Speedup = 2

$$SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1-P) + \frac{P}{N}}$$

$$SP(\infty) = \frac{1}{(1-P)}$$

Amdahl's Law

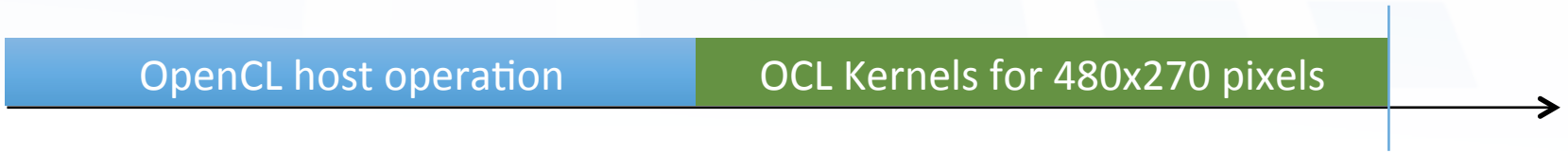


$$SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1-P) + \frac{P}{N}}$$

$$SP(\infty) = \frac{1}{(1-P)}$$

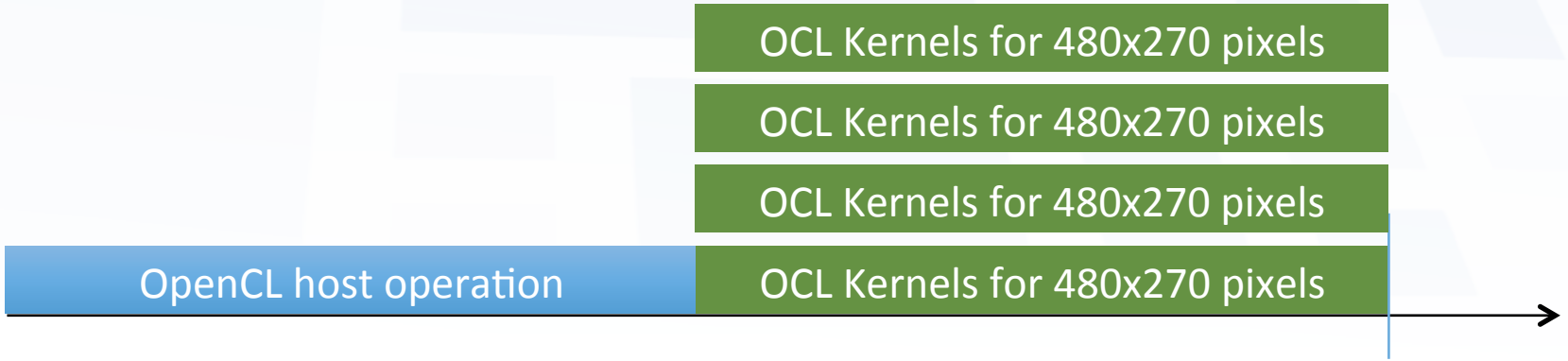
Gustafson's Law

- Workload scales with computing facilities
  - (50% parallelizable P=0.5)
  - On a sequential processor (single core) takes 1 unit of time to complete workload W(1) designed with a single core in mind



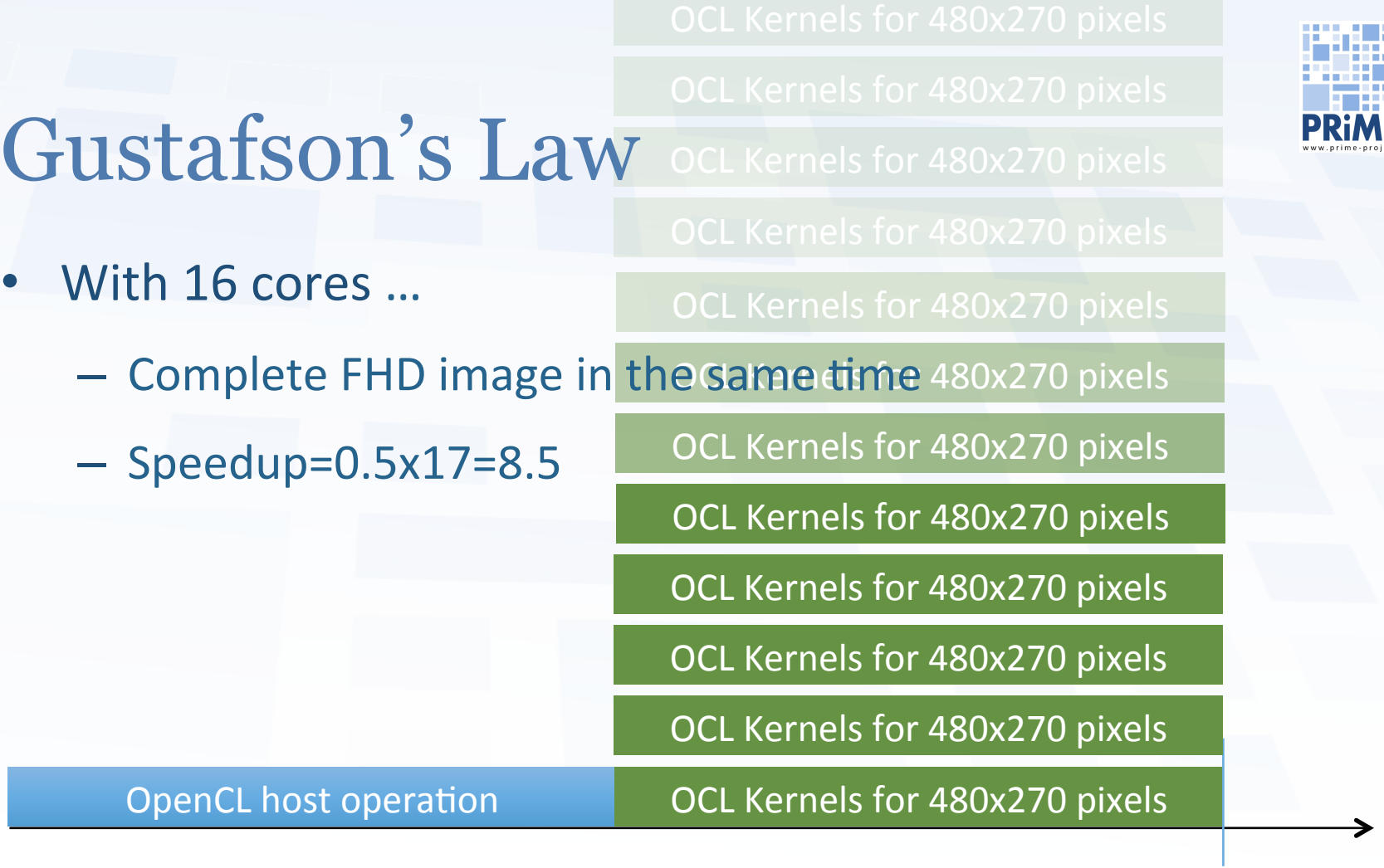
Gustafson's Law

- With four cores …
  - Complete 960x540 image in the same time
  - 0.5x5=2.5 times the workload (speedup=2.5)



Gustafson's Law

- With 16 cores …
  - Complete FHD image in the same time
  - Speedup=0.5x17=8.5



Gustafson's Law

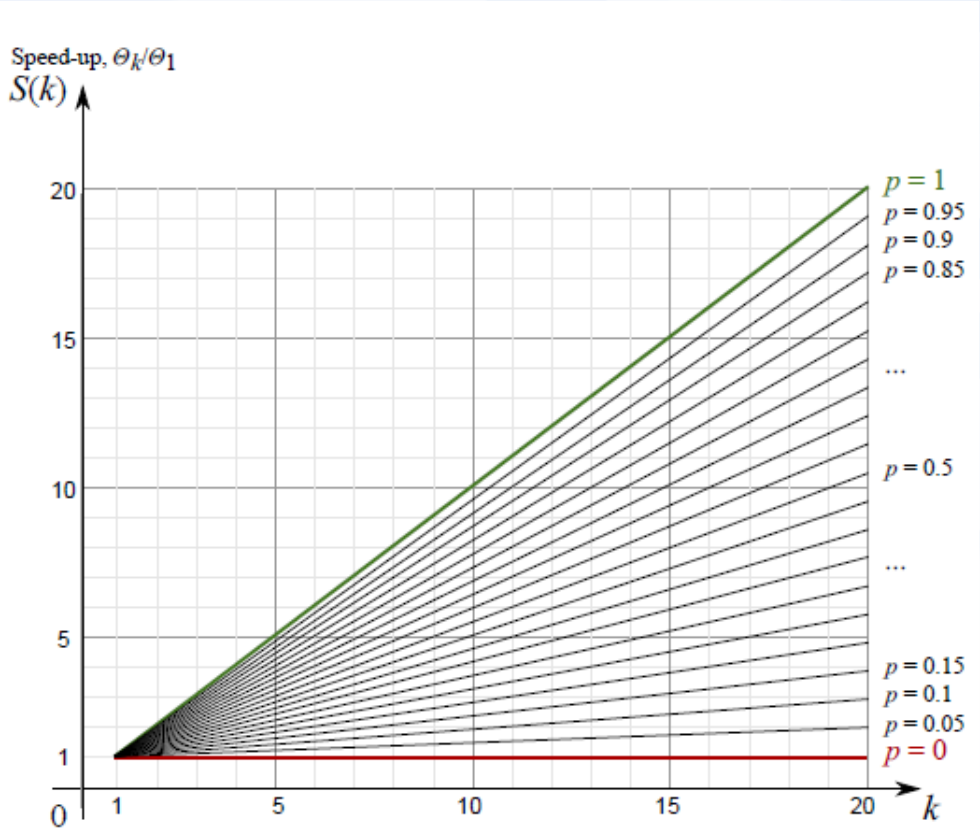
- Speedup is the ratio between the improved workload and the workload before improvement
  - Calculated at fixed time

$$W(1) = (1 - P)W + PW$$

$$W(N) = (1 - P)W + PNW$$

$$SP(N) = \frac{W(N)}{W(1)} = (1 - P) + PN$$

Gustafson's Law



$$W(1) = (1 - P)W + PW$$

$$W(N) = (1 - P)W + PNW$$

$$SP(N) = \frac{W(N)}{W(1)} = (1 - P) + PN$$

Sun-Ni Law

- Memory-bound speedup model
  - Parallel workload per core restricted by memory structure (multi-level caches, shared memory/interfaces, etc.)
  - One core's workload capability restricted by M – the memory of one core, N cores' workload capability restricted by NxM
  - For the P part:

$$W(1) = G(M)$$

$$W(N) = G(N \times M) = G(N \times G^{-1}(W(1)))$$

Sun-Ni Law

- Multi-core speedup is derived thusly:

$$W(N) = (1-P)W(1) + P \times G(N \times G^{-1}(W(1)))$$

$$W(1) = (1-P)W(1) + P \times G(M)$$

$$W(1) = (1-P)W(1) + \frac{P \times G(N \times G^{-1}(W(1)))}{N}$$

$$SP(N) = \frac{W(N)}{W(1)} = \frac{(1-P)W(1) + P \times G(N \times G^{-1}(W(1)))}{(1-P)W(1) + \frac{P \times G(N \times G^{-1}(W(1)))}{N}}$$

Sun-Ni Law

- Trying to remove W(1):

$$SP(N) = \frac{W(N)}{W(1)} = \frac{(1-P)W(1) + P \times G(N \times G^{-1}(W(1)))}{(1-P)W(1) + \frac{P \times G(N \times G^{-1}(W(1)))}{N}}$$

$$if\ G(x) = ax^b\ with\ rational\ a\ and\ b$$

$$G(N \times x) = N^b \times ax^b = N^b \times G(x), then$$

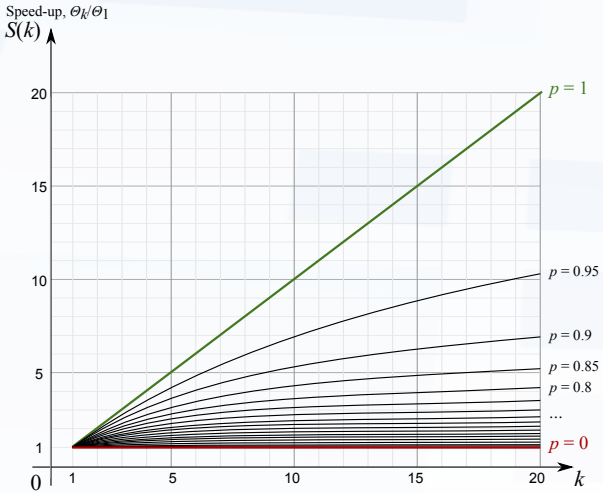
$$G\left(N \times G^{-1}\left(W(1)\right)\right) = N^b \times G(G^{-1}\left(W(1)\right) = N^b \times W(1)$$

$$SP(N) = \frac{(1-P) + P \times g(N)}{(1-P) + \frac{P \times g(N)}{N}}, with\ g(N) = N^b$$

Sun-Ni Law

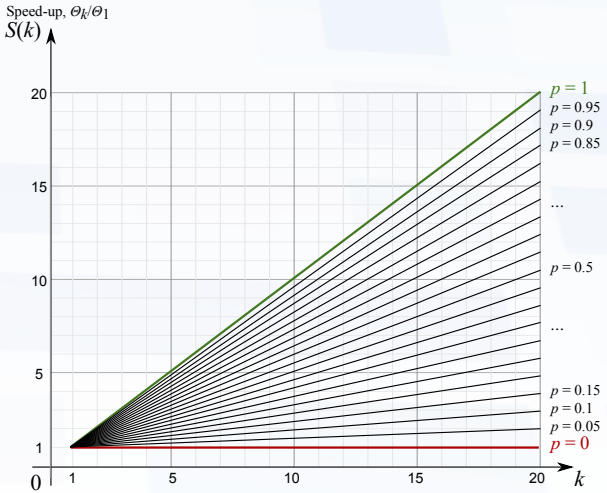
- Depending on g(N)
  - Sub-linear scaling (Amdahl's if g(N)=1)
  - Linear scaling (Gustafson's if g(N)=N)
  - Super-linear scaling (if g(N)>N)
- If you had more memory than cores, and the problem is memory-bound, you can scale to higher speedup than what your cores allow for compute-bound problems

Comparing the three models



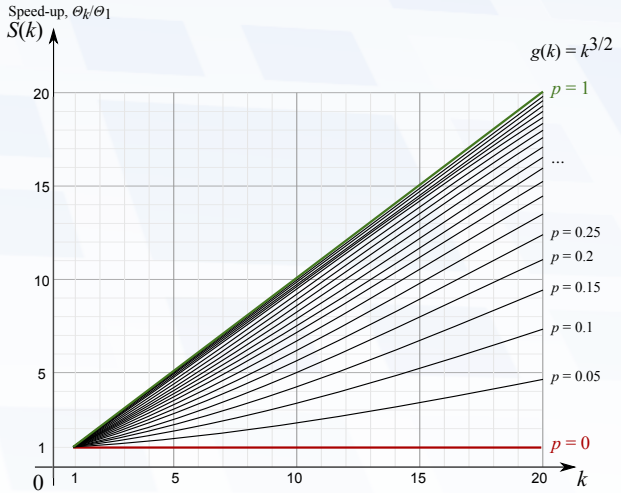
Amdahl's Law

$$S(k) = \frac{1}{(1-p) + \frac{p}{k}}$$ 1



Gustafson's Model

$$S(k) = (1-p) + pk$$ 2



Sun and Ni's Model

$$S(k) = \frac{(1-p) + pg(k)}{(1-p) + \frac{pg(k)}{k}}$$ 3

Outline

- Existing speedup models
- Motivation
- Extended heterogeneous speedup models
- Power consumption models
- Power and energy normalized speedup
- Experimental results and cross validation
- Conclusions

Motivation

- Extend to a higher degree of core heterogeneity
- Extend to power/energy/efficiency and cover modes like dynamic voltage and frequency scaling (DVFS)
- Potential applications in run-time management systems of parallel systems

Existing Speedup Models and the Extended Model

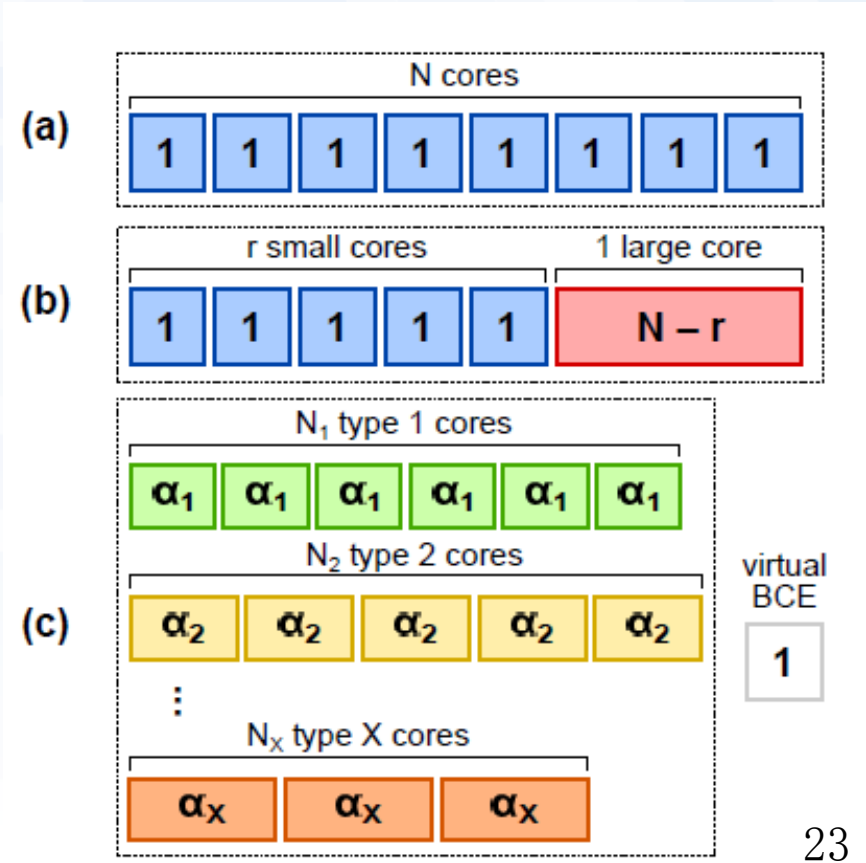
	Homogeneity	Heterogeneity	Power	Amdahl	Gustafson	Sun and Ni
Amdahl	Yes	No	No	Yes	No	No
Gustafson	Yes	No	No	Yes	Yes	No
Sun and Ni	Yes	No	No	Yes	Yes	Yes
Hill-Marty	Yes	Simple	No	Yes	No	No
Hao and Xie	Yes	Simple	No	Yes	Yes	Yes
Woo and Lee	Yes	Simple	Yes	Yes	No	No
Sun and Chen	Yes	No	No	Yes	Yes	Yes
Extended Model	**Yes**	**Normal**	**Yes**	**Yes**	**Yes**	**Yes**

Outline

- Existing speedup models
- Motivation
- Extended heterogeneous speedup models
- Power consumption models
- Power and energy normalized speedup
- Experimental results and cross validation
- Conclusions

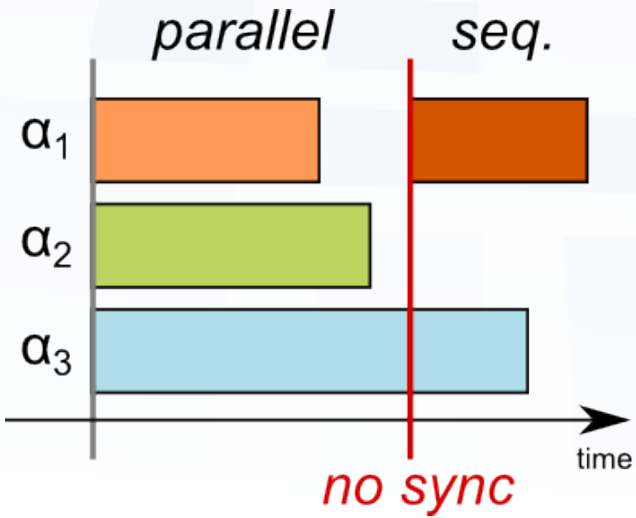
Heterogeneity

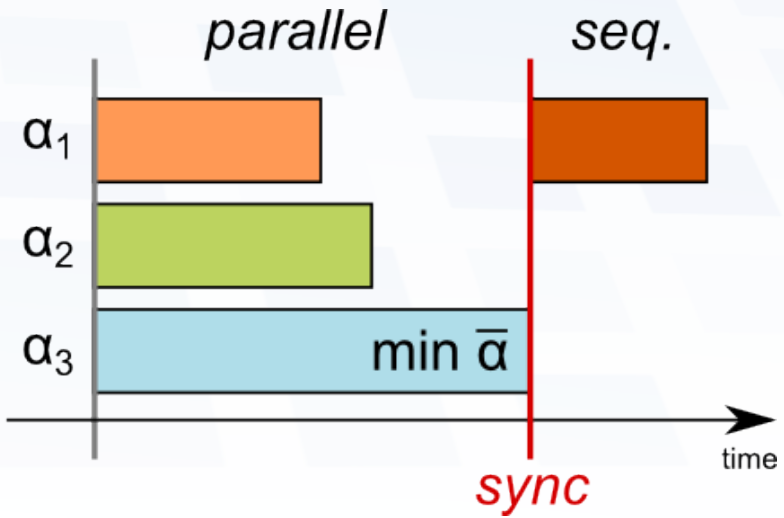
- Existing heterogeneity include ‘asymmetric’ and ‘dynamic’ structures (b) ☹
- We extend to cover the normal form of core heterogeneity ☺
- Still iso-ISA and not fully general ☹



Heterogeneity

- Parallel computation may not all finish together (Amdahl's)





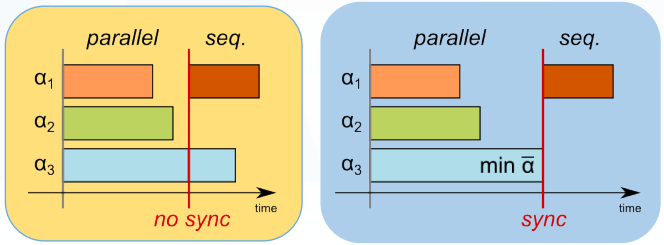
BCE performance equivalence

- Calculating the performance equivalent number of BCEs
  - Based on the slowest (last to finish) core

$$N_\alpha = \min \bar{\alpha} \cdot \sum_{i=1}^{X} N_i$$

$\max \bar{\alpha}$, $\text{avr}\ \bar{\alpha}$, $\text{opt}\ \bar{\alpha}$ also possible

$$\bar{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_X\}$$

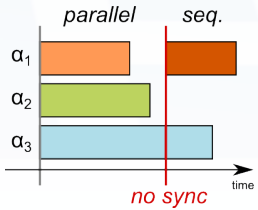


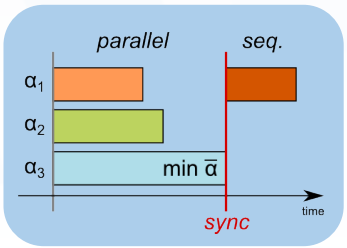
Speedup extension Amdahl's

- Calculating the performance equivalent number of BCEs
  - Based on the slowest (last to finish) core

$$N_\alpha = \min \bar{\alpha} \cdot \sum_{i=1}^{X} N_i$$

$$SP(\bar{N}) = \frac{1}{\frac{(1-P)}{\alpha_X} + \frac{P}{N_\alpha}}, (Amdahl's)$$





Speedup extension Amdahl's

- Calculating the performance equivalent number of BCEs
  - Based on the slowest (last to finish) core

$$N_\alpha = \min \bar{\alpha} \cdot \sum_{i=1}^{X} N_i$$

Now in vector space

$$SP(\bar{N}) = \frac{1}{\frac{(1-P)}{\alpha_X} + \frac{P}{N_\alpha}}, (Amdahl's)$$

Sequential on fastest core, if α_X is fastest

Parallel synced to the slowest, in case of min α

$$SP(N) = \frac{T(1)}{T(N)} = \frac{1}{(1-P) + \frac{P}{N}}$$

Speedup extension Gustafson's

- Gustafson's speedup model extension
  - Again assuming sequential on a type X core and parallel on N_α

$$SP(\overline{N}) = (1 - P)\alpha_X + PN_\alpha$$

Sequential on fastest core, if α_X is fastest

Parallel synced to the slowest, in case of min α

$SP(N) = (1 - P) + PN$

Speedup extension Sun-Ni's

- Extending Sun-Ni's model
  - Again assuming sequential on a type X core and parallel on N_α

Memory bound function for all cores

$$SP(\bar{N}) = \frac{(1-P) + Pg(\bar{N})}{(1-P)\alpha_X + \frac{Pg(\bar{N})}{N_\alpha}}$$

Sequential on fastest core, if α_X is fastest

Parallel synced to the slowest, in case of min α

$$SP(N) = \frac{(1-P) + Pg(N)}{(1-P) + \frac{Pg(N)}{N}}$$

Reduces to extended Amdahl's and Gustafson's as expected

Outline

- Existing speedup models
- Motivation
- Extended heterogeneous speedup models
- Power consumption models
- Power and energy normalized speedup
- Experimental results and cross validation
- Conclusions

Power

- Divide power into effective and idle

$$W_{total} = W(N) + W_{idle}$$

Effective power: power used by workload

$$W(N) = \frac{W_S T_S(N) + W_P T_P(N)}{T_S(N) + T_P(N)}$$

$W_S = \beta_X W_1$, W_1 is the power of one BCE

$$W_P = W_1 \sum_{i=1}^{X} \beta_i N_i = N_\beta W_1$$

Idle power includes both static power and active power that's not used by workload

$W_{idle} = N_i \cdot W_i$

when N_i BCEs are idle

Effective power

- The βs are similar to the αs, but pertain to power
  - An ith-type core consumes $\beta_i W_1$ power and has α_i speed (speed of one core is 1 – for throughput, we deal with speedup, for power, we deal with wattage and not ratios)
  - N_β is the power-equivalent number of BCEs
  - For synchronizing on the slowest core:

$$N_\beta = \min \bar{\alpha} \cdot \sum_{i=1}^{X} \frac{N_i \beta_i}{\alpha_i}$$

Effective power

- Effective power formulas have been derived for all three types of models/laws
  - Can be viewed as results of ‘power scaling’ with PS functions:

$$PS\left(\overline{N}\right) = \frac{\beta_X}{\alpha_X} \cdot (1 - P) + \frac{N_\beta}{N_\alpha} \cdot P$$

$$PS\left(\overline{N}\right) = \frac{\beta_X \cdot (1 - P) + N_\beta \cdot P}{\alpha_X \cdot (1 - P) + N_\alpha \cdot P}$$

$$PS\left(\overline{N}\right) = \frac{\frac{\beta_X}{\alpha_X} \cdot (1 - P) + \frac{N_\beta}{N_\alpha} \cdot P \cdot g\left(\overline{N}\right)}{(1 - P) + P \cdot g\left(\overline{N}\right)}$$

$$W(\overline{N}) = PS(\overline{N}) \cdot SP(\overline{N}) \cdot W_1$$

With $\alpha_i = \beta_i = 1, \forall i$, and $\overline{N} = N$
all models transform to homogeneous forms

Efficiency

- Power-normalized performance
  - IPS/Watt
- Energy per instruction
  - Joules/Instruction
- Energy-normalized performance
  - IPS/Joule
- All models in the paper

Outline

- Existing speedup models
- Motivation
- Extended heterogeneous speedup models
- Power consumption models
- Power and energy normalized speedup
- Experimental results and cross validation
- Conclusions

Experimental platform

Exynos 5422 Application Processor			
CPU			
Cortex-A15 Quad (2.0 GHz)		Cortex-A7 Quad (1.4 GHz)	
Cortex-A15 32 KB instruction cache 32 KB data cache VFPv4	Cortex-A15 32 KB instruction cache 32 KB data cache VFPv4	Cortex-A7 32 KB instruction cache 32 KB data cache VFPv4	Cortex-A7 32 KB instruction cache 32 KB data cache VFPv4
Cortex-A15 32 KB instruction cache 32 KB data cache VFPv4	Cortex-A15 32 KB instruction cache 32 KB data cache VFPv4	Cortex-A7 32 KB instruction cache 32 KB data cache VFPv4	Cortex-A7 32 KB instruction cache 32 KB data cache VFPv4
2 MB Level 2 Cache with ECC		512 KB Level 2 Cache	
GPU Mali-T628 MP6 (600 MHz)		DRAM LPDDR3 (933 MHz) 14.9 GBytes/s	

System characterization

- Build parameters through experimentation
  - W_{A7}, W_{A15}, W_{idle} (for the 'whole' – did not try differentiating different W_i – cores not turned off even in N_{A7}=0 and N_{A15}=0 cases)
  - α_{A7}, α_{A15}, β_{A7}, β_{A15}
  - May be different for different apps/computations
  - CPU-heavy tasks mainly experimented in this initial study, min α scheduling
  - log, sqrt, and integer arithmetic tested

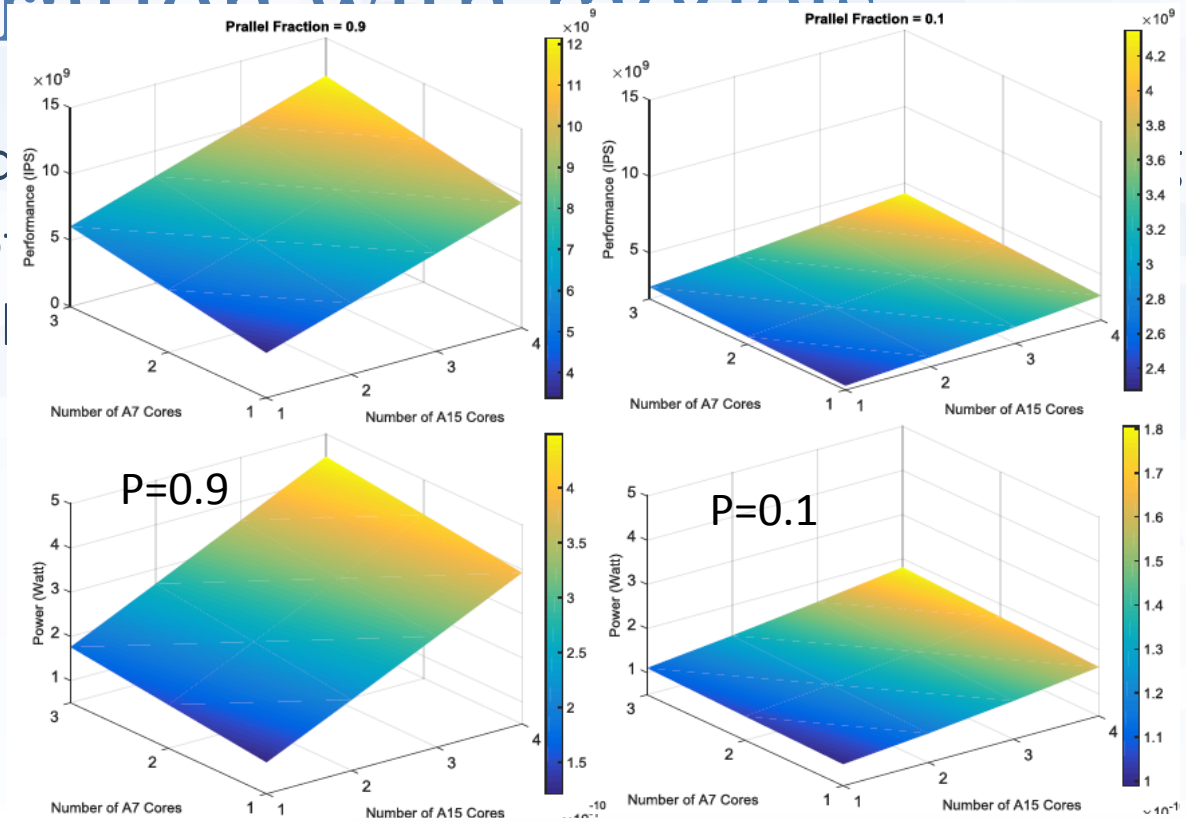
Exploration with models

- Run models with various execution scenarios to investigate the effects of P, DVFS, core scaling, etc. (large database available from technical report, some example data in the paper)

http://async.org.uk/tech-reports/NCL-EEE-MICRO-TR-2016-198.pdf

Exploration with models

- Run moc ate the effects o lable from technical



http://async.org.uk/tech-reports/NCL-EEE-MICRO-TR-2016-198.pdf

Cross-validation

				time, ms	speedup			average total power, W		
bench	P	N_{A7}	N_{A15}	measured	predicted	measured	error	predicted	measured	error
sqrt	0.3	3	0	59992	1.2500	1.2505	0.04%	0.6622	0.6686	0.96%
sqrt	0.3	0	4	61911	1.2116	1.2117	0.01%	1.1119	1.0993	1.15%
sqrt	0.3	2	2	61910	1.2116	1.2118	0.01%	1.0438	1.0312	1.22%
sqrt	0.3	3	4	59359	1.2641	1.2638	0.02%	1.0769	1.0666	0.97%
sqrt	0.9	3	0	29988	2.5000	2.5017	0.07%	0.8122	0.8042	0.99%
sqrt	0.9	0	4	25977	2.8893	2.8879	0.05%	1.9424	1.9252	0.89%
sqrt	0.9	2	2	25961	2.8893	2.8897	0.02%	1.4548	1.4239	2.17%
sqrt	0.9	3	4	18300	4.1082	4.0995	0.21%	1.9515	1.9403	0.58%
int	0.9	3	0	31705	2.5000	2.5021	0.08%	0.8265	0.8305	0.49%
int	0.9	0	4	20823	3.8112	3.8097	0.04%	1.9457	1.9351	0.55%
int	0.9	2	2	24264	3.2708	3.2694	0.04%	1.3739	1.3537	1.49%
int	0.9	3	4	16637	4.7777	4.7682	0.20%	1.8478	1.8117	1.99%
log	0.9	3	0	25118	2.5000	2.5053	0.21%	0.8900	0.8925	0.29%
log	0.9	0	4	11580	5.4219	5.4341	0.22%	2.2497	2.2135	1.64%
log	0.9	2	2	17722	3.5492	3.5508	0.04%	1.3791	1.3615	1.29%
log	0.9	3	4	11690	5.3960	5.3830	0.24%	1.8889	1.8570	1.72%

Max 2.17%

Max 0.24%

Outline

- Existing speedup models
- Motivation
- Extended heterogeneous speedup models
- Power consumption models
- Power and energy normalized speedup
- Experimental results and cross validation
- Conclusions

Conclusions

- Extended popular parallelization speedup models to cover a wider range of iso-ISA (or coefficient-equivalent ISA) heterogeneity
- Extended power models and efficiency models
- First cross-validation study successful
- Need to investigate even wider scopes of heterogeneity
- Need to study other speedup models (e.g. Downey's)
- Need to investigate realistic memory subsystems (cache misses)
- Need to explore using these models for run-time management